IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 10/772,518	§	Examiner:	Chen, Qing
Filed: February 5, 2004	§	Group/Art Unit:	2191
Inventor(s):	§	Atty. Dkt. No:	5150-38605
Robert E. Dye, Darshan Shah,	§		
Steve Rogers, Greg Richardson,	§		
Dean A. Luick	§		
Title: GRAPHICAL	§		
PROGRAMMING	§		
SYSTEM WITH BLOCK	§		
DIAGRAM EXECUTION	§		
AND DISTRIBUTED	§		
USER INTERFACE	§		
DISPLAY	§		
	§		

APPEAL BRIEF

Box: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Notice of Appeal filed August 24, 2009, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 59-104 are pending in the case. Claims 59-104 stand rejected under 35 U.S.C. § 102(b) and are the subject of this appeal. A copy of claims 59-104, incorporating entered amendments, as on appeal, is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

Subsequent to the rejection in the Final Office Action of June 4, 2009, the claims were amended in the Response After Final Rejection of July 6, 2009, and in the Supplemental Response to Final Office Action of August 4, 2009. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The present invention relates to the field of graphical programming, and more particularly to a graphical programming system with block diagram execution and a distributed user interface that presents both a graphical user interface associated with the block diagram as well as the block diagram itself.

Independent claim 59 recites a computer accessible memory medium that stores program instructions executable to perform the following: A network connection is established with a client computer system over a network, and user input received from the client computer system indicating a graphical program for execution. *See, e.g., p.22:21-p.23:29; Figure 7:450/452/454.* The graphical program includes a block diagram that includes a plurality of interconnected function icons representing graphical data flow of a desired function. *See, e.g., p.4:1-26, and p.20:1-p.21:21; Figure 8B.* The graphical program is executed, which includes executing the block diagram. *See, e.g., p.3:27-29.* Information describing a user interface of the graphical program is sent over the network to the client computer system after establishing the network connection with the client computer system, where the information regarding the user interface is useable by the client computer system to display the user interface on the client computer system. *See, e.g., p.24:1-23; Figure 7:458.* Information regarding the block diagram of the graphical program is sent over the network to the client computer system after establishing the network connection with the client computer system, where the information regarding the block diagram is useable by the client computer system to display the block diagram on the client computer system. *See, e.g., p.26:21-p.27:2.* The user interface is operable to facilitate interaction between a user and the graphical program over the network. *See, e.g., p.25:11-24; Figure 7:460/462.*

Independent claim 73 is directed to a method for executing a graphical program on a first computer and providing a user interface of the graphical program for display on a second computer. *See, e.g., p.1:22-29.* The graphical program is executed on the first computer, where the graphical program includes a block diagram that comprises a

plurality of interconnected function icons representing graphical data flow of a desired function, where the first computer and the second computer are connected over a network, and where executing the graphical program includes executing the block diagram. *See, e.g., p.4:1-26, p.20:1-p.21:21, p.3:27-29; Figure 8B.* Information describing the user interface of the graphical program is provided to the second computer over the network during the executing, where the information describing the user interface is useable by the second computer to display the user interface of the graphical program on the second computer. *See, e.g., p.24:1-23; Figure 7:458.* Information regarding the block diagram of the graphical program is provided by the first computer to the second computer over the network, where the information regarding the block diagram is useable by the second computer to display the block diagram on the second computer. *See, e.g., p.26:21-p.27:2.* The user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer. *See, e.g., p.25:11-24; Figure 7:460/462.*

Independent claim 81 is directed to a system for executing a graphical program. *See, e.g., p.1:22-29.* The system includes a first computer, including a processor, and a memory, coupled to the processor. *See, e.g., Figure 3; p.18:12-28.* The first computer is operable to couple to a network. *See, e.g., p.13:3-7.* The memory stores a graphical program that includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function. *See, e.g., p.13:16-20; Figure 8B.* The first computer is operable to execute the graphical program and provide information describing a user interface of the graphical program over the network to a second computer during said executing, where executing the graphical program includes executing the block diagram, where the information describing the user interface over the network is useable by the second computer to display the user interface of the graphical program. *See, e.g., p.23:27-29, See, e.g., p.24:1-23; Figure 7:458.* The user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer. *See, e.g., p.25:11-24; Figure 7:460/462.* The first computer is operable to provide information regarding the block diagram of the graphical program over the network to the second computer, where the information

regarding the block diagram is useable by the second computer to display the block diagram on the second computer. *See, e.g., p.26:21-p.27:2.*

Independent claim 82 is directed to a computer accessible memory medium that stores program instructions executable to perform the following: User input is received at a first computer indicating a graphical program, where the graphical program is stored on a server computer, where the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function. *See, e.g., p.20:1-p.21:21, p.22:21-p.23:29, p.4:1-26; Figure 7:450/452/454; Figure 8B.* The user input indicating the graphical program is provided over a network to the server computer. *See, e.g., p.23:3-27.* Information describing a user interface of the graphical program is received from the server computer over the network during execution of the graphical program on the server computer. *See, e.g., p.24:1-23; Figure 7:458.* Information regarding the block diagram of the graphical program is received from the server computer over the network. *See, e.g., p.26:21-p.27:2.* The user interface of the graphical program is displayed at the first computer based on the information describing a user interface. *See, e.g., p.24:5-14.* The block diagram is displayed at the first computer based on the information regarding the block diagram. *See, e.g., p.26:21-p.27:2.* The user interface facilitates interaction between a user and the graphical program executing on the server computer. *See, e.g., p.25:11-24; Figure 7:460/462.*

Independent claim 96 is directed to a method for displaying a graphical user interface and block diagram of a graphical program on a second computer in response to execution of the graphical program on a first computer. *See, e.g., p.1:22-29.* User input is received to the second computer, where the user input indicates the graphical program on the first computer, where the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function, and where the first computer and the second computer are connected over a network. *See, e.g., p.13:3-7, p.20:1-p.21:21, p.22:21-p.23:29, p.4:1-26; Figure 7:450/452/454; Figure 8B.* Information describing a graphical user interface of the

graphical program is received at the second computer from the first computer over the network during execution of the graphical program on the first computer. *See, e.g., p.24:1-23; Figure 7:458.* Information regarding the block diagram of the graphical program at the second computer from the first computer over the network. *See, e.g., p.26:21-p.27:2.* The graphical user interface of the graphical program is displayed on the second computer based on the information describing the graphical user interface. *See, e.g., p.24:5-14.* The block diagram is displayed on the second computer based on the information regarding the block diagram. *See, e.g., p.26:21-p.27:2.* The graphical user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer. *See, e.g., p.25:11-24; Figure 7:460/462.*

Independent claim 104 is directed to a system that includes a processor, a memory, coupled to the processor, and a network port operable to couple to a network. *See, e.g., Figure 1; p.13:3-7, Figure 3; p.18:12-28.* The memory medium stores program instructions executable by the processor to perform the following: user input specifying a graphical program is received, where the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function. *See, e.g., p.13:3-7, p.20:1-p.21:21, p.22:21-p.23:29, p.4:1-26; Figure 7:450/452/454; Figure 8B.* The user input specifying the graphical program is provided over a network to a server computer. *See, e.g., p.23:3-27.* Information describing a user interface of the graphical program is received from the server computer over the network during execution of the graphical program on the server computer. *See, e.g., p.24:1-23; Figure 7:458.* Information regarding the block diagram of the graphical program is received from the server computer over the network. *See, e.g., p.26:21-p.27:2.* The user interface of the graphical program is displayed at the first computer based on the information describing a user interface. *See, e.g., p.24:5-14.* The block diagram is displayed at the first computer based on the information regarding the block diagram. *See, e.g., p.26:21-p.27:2.* The user interface facilitates interaction between a user and the graphical program executing on the server computer. *See, e.g., p.25:11-24; Figure 7:460/462.*

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 59-104 were rejected under 35 U.S.C. 103(a) as being unpatentable over US Pat. No. 5,801,689 ("Huntsman), in view of US Pat. No. 4,901,221 ("Kodosky").

VII. ARGUMENT

Ground of Rejection

Claims 59-104 were rejected under 35 U.S.C. 103(a) as being unpatentable over US Pat. No. 5,801,689 ("Huntsman), in view of US Pat. No. 4,901,221 ("Kodosky"). Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 59, 61, 62, 63, 64, 65, 66, 67, 70, 71, 72, 73, 76, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96, 97, 98, 99, 101, 102, 103, 104:

Independent claim 59 is separately patentable because the cited reference does not teach or suggest the limitations recited in this claim. For example, Appellant respectfully submits that the cited art fails to teach or suggest at least: **send information regarding the block diagram of the graphical program over the network to the client computer system after establishing the network connection with the client computer system, wherein the information regarding the block diagram is useable by the client computer system to display the block diagram on a client computer system**, as recited in claim 59.

Appellant respectfully submits that the combination of Huntsman and Kodosky fails to teach or suggest these features.

The Advisory Action continues to argue that any program with a graphical user interface (GUI) is a "graphical program", in direct contradiction to the claim and specification. This is incorrect and improper. Note, for example, that claim 59 states: "the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function, and wherein said executing the graphical program comprises executing the block diagram". Additionally, claim 59 clearly refers to the graphical program's block diagram and user interface as separate and distinct parts of the graphical program.

It is improper for the Examiner to ignore the definition of "block diagram" contained in the claim ("a block diagram that comprises a plurality of interconnected

function icons representing graphical data flow of a desired function"), in an attempt to blur the distinctions between text-based programs, such as Huntsman's MS-Windows programs, and Appellant's graphical programs. Applicant respectfully submits that one of skill in the programming arts readily understands that text-based programs are written in text-based programming languages, e.g., FORTRAN, PASCAL, C, C++, etc., i.e., have textual source code, and may or may not include GUIs, whereas graphical programs have graphical source code, i.e., a block diagram comprising interconnected graphical program icons or nodes. For example, the cited MS-Windows and Windows Explorer are textual programs, even though they implement GUIs, and the cited folder and file icons are GUI elements, not graphical source code of the programs, and certainly not interconnected icons of a block diagram of a graphical program as recited in claim 59. In other words, the fact that a program's GUI includes "icons, panels, and windows" does not make the program a "graphical program" as defined in the present claim. Thus, these cited icons and other GUI elements are not germane to these claimed features.

The Advisory Action further argues that "the GIF file is a color image of the executing GUI based program", and that "the GIF file represents the GUI features and the graphical program features of the executing GUI-based program". This is incorrect. The GIF file is directed at the GUI of a text-based program, not the program (source code) itself. As the first paragraph of the Detailed Description makes clear, Huntsman's remote control system has two main components: a GUI-screen-to-hypertext convertor, and a hypertext-to-GUI response means, both directed to translating a textual program's GUI to or from hypertext, e.g., translating a Windows (text-based) program's GUI screen to HTML and GIF data files, and vice versa, respectively, and thus, the GIF file only represents a text-based program's GUI, and is not germane to the claimed graphical program block diagram.

Appellant respectfully requests that the term "graphical program" be interpreted as clearly recited in the Application and claims.

Regarding cited col.9:31-50 of Huntsman, Appellant respectfully notes that the Examiner cites Huntsman's sending of GUI information ("the GIF image file containing the screen image of the GUI on the first computer") with respect to *both* sending information describing a user interface of the graphical program over a network to the

client computer system, *and* sending information regarding the block diagram of the graphical program over the network to the client computer system. This is incorrect, and improper, as it ignores key distinctions of elements recited in the claim. In other words, Huntsman's sending of GUI information cannot properly be cited as teaching both sending information regarding the graphical user interface of the graphical program, *and* sending information regarding the block diagram of the graphical program.

Per the citation, Huntsman's GIF file, as referenced by the REMOTE.HTM, presents the screen image of the client (first) computer, and REMOTE.HTM contains appropriate HTML references to the GIF file so that the GIF file will be displayed as a clickable image, and thus appears to be a GUI. As one of skill in the programming arts would readily understand, such a GUI is not a graphical program, nor, more specifically, a block diagram, as clearly defined in claim 59, where the graphical program includes the block diagram, and where the block diagram includes a plurality of interconnected function icons representing graphical data flow of a desired function, and where executing the graphical program includes executing the block diagram. Nor does Huntsman ever even mention or hint at a graphical program as clearly defined in the claim, and specification.

The Office Action asserts that Huntsman discloses "send information regarding the graphical program over the network, wherein the information regarding the graphical program is useable by the client", citing the same GUI information that was cited against Appellant's claimed user interface information. The Office Action then admits that Huntsman fails to teach the specific feature that the graphical program includes the block diagram, where the block diagram includes a plurality of interconnected function icons representing graphical data flow of a desired function, and where executing the graphical program includes executing the block diagram. The Office Action then asserts that Kodosky remedies this deficiency, citing Kodosky's block diagrams.

In other words, regarding Huntsman, the Examiner appears to assert a rough equivalence between Huntsman's "GUI" and a "graphical program" (which is incorrect, as explained above), then appears to assert a rough equivalence between "GUI" and Kodosky's block diagram, thereby blurring the distinction between a GUI and a block diagram (which is graphical program code), which is improper. Appellant submits that it

is improper to assemble disparate terms from the cited art in an attempt to produce Appellant's claimed invention as represented in claim 59.

Thus, the Examiner's assertion that Huntsman discloses "send information regarding the graphical program over the network" is incorrect, at least because the GUI information Huntsman sends is not regarding a graphical program at all, and because the GUI information was characterized by the Examiner as Appellant's claimed information regarding a user interface *for* a graphical program, which is quite different.

Kodosky does not mention or even hint at sending information regarding a block diagram of a graphical program over a network to a client computer system, where the information regarding the block diagram is useable by the client computer system to display the block diagram. Kodosky discusses graphical programs, including a block diagram, but nowhere discloses sending information regarding a block diagram over a network for use in displaying the block diagram. Appellant notes that the citation (col.8:8-23) simply describes a graphical program's block diagram and front panel (GUI), but says nothing regarding sending information regarding a block diagram over a network for display of the block diagram. Said another way, Kodosky discloses graphical programs, but nowhere teaches sending information regarding a block diagram of a graphical program over a network to a client system for display.

Appellant submits that Huntsman's sending of GUI information and Kodosky's graphical program in combination do not produce this limitation of claim 59. More specifically, if one were to combine Kodosky and Hunstman, the result would NOT involve sending graphical source code over a network to a remote computer. This is because neither reference involves transmitting source code to a remote computer. Hunstman only transmits a graphical user interface, and does not send any of its textual source code. Kodosky nowhere teaches transmission of its graphical source code (block diagram) over a network. Thus the combination lacks a key element of claim 59, and hence is improper.

Moreover, Appellant respectfully submits that one of skill in the art would not combine Huntsman and Kodosky in the manner proposed by the Examiner. As noted above, the Huntsman reference does not teach sending textual code across a network for display to a user. Appellant notes that there is no reason a client user of Huntsman's

system would benefit from seeing pages and pages of textual source code. Even an expert programmer would likely to have significant difficulty understanding program functionality based on voluminous text-based program source code. Thus a typical program *user* would have no use for such textual source code. Thus, Huntsman actually teaches away from claim 1.

In direct contrast, graphical source code, i.e., Applicant's claimed block diagram, graphically indicates the functionality of the graphical program, and thus there is benefit to sending the block diagram over a network to a client system for viewing by the user.

Appellant again notes that combining Huntsman and Kodosky would simply NOT produce this functionality, at least because Huntsman teaches sending GUI information, *not* source code (which, being textual, is not useful to typical users), and Kodosky doesn't teach sending either GUI information or the block diagram.

Thus, the alleged combination of Huntsman and Kodosky is invalid, and does not teach these claimed features.

Applicant now presents rebuttals to various of the Examiner's Response to Arguments:

In the Response to Arguments of the Office Action, the Examiner argues that "one or ordinary skill in the art would readily comprehend that a MS-Windows program is a graphical program (e.g., panels and windows) that is manipulated by a user via a GUI (e.g., buttons and menus)." Again, this is incorrect and improper, since the cited MS-Windows programs do not meet the definition of graphical program as clearly stated in both the claims and the Specification (see quotes above).

In the Response to Arguments, the Examiner further argues that Kodosky is only relied on for "information regarding the block diagram of the graphical program", and that "Huntsman clearly discloses sending information regarding the graphical program over a network to a client system for display…". Appellant respectfully notes that if Kodosky is relied on only for "information regarding the block diagram of the graphical program", that since Huntsman only discloses sending GUI information for a textual program over a network, then the combination does not, and cannot, disclose sending information describing a user interface for a graphical program over a network, *and* sending information describing a block diagram for the graphical program over the

network, as claimed. Moreover, as explained repeatedly, Huntsman's programs are not graphical programs, and Huntsman's GUIs are not user interfaces of graphical programs. Appellant respectfully requests that the Examiner give proper weight and consideration to the terms in Appellant's claims, and submits that if such consideration is given, it will be clear that Kodosky and Huntsman do not, and cannot, teach these claimed features.

In the Response to Arguments, the Examiner asserts that Appellant's arguments fail to comply with 37 CFR 1.111(b) because "they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Appellant respectfully disagrees, noting that Appellant has simply explained how the Examiner has failed to show that the cited art discloses the subject matter of Appellant's claims, and has further explained that the references do not and cannot teach key limitations of the claims.

Regarding the Examiners assertion that Appellant has improperly only attacked the references individually, Appellant respectfully disagrees. Appellant has rebutted specific assertions by the Examiner regarding what each individual references teaches, particularly Huntsman, and respectfully notes that the Examiner's combination argument relies on these assertions, and so fails when the assertions are incorrect. Appellant has also addressed the fact that in combination, these references also fail to produce Appellant's claimed embodiments.

The Office Action further asserts that Huntsman's Abstract teaches a graphical program as recited in the independent claims. The cited Abstract reads thusly:

> A remote control system for remotely controlling a Microsoft Windows or other GUI-based first computer from a second computer over the internet using only a standard world-wide-web hypertext browser on the second computer. The second controlling computer may be dissimilar from the first controlled computer user interface, underlying operating system, and hardware architecture.

As may be seen, nowhere does this citation mention, or even hint at, a graphical program as defined in the claims, specifically, "wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing

graphical data flow of a desired function", nor as defined in the specification. Thus, the Office Action's assertion is incorrect.

The Examiner continues to argue that Huntsman and Kodosky in combination disclose sending information regarding a block diagram over a network to a client system. However, as explained above at length, this is not the case. Huntsman discloses sending information regarding a Windows computer screen, e.g., a GUI, over a network, and Kodosky discloses a block diagram, but neither reference teaches, or even hints at the desirability of sending information regarding a block diagram over a network. Thus, even in combination, these references would not produce this claimed feature.

Appellant respectfully submits that Appellant has properly addressed the Examiner's assertions, explaining carefully and at length that Huntsman's MS-Windows programs *are not graphical programs as defined in the claims* for at least the reason that they do not include a block diagram that includes a plurality of interconnected function icons representing graphical data flow of a desired function, and that executing the graphical program includes executing the block diagram. The Examiner simply seems to be ignoring or dismissing these technical features. As another example, Huntsman never even mentions the term "edit", nor describes editing any programs at all, particularly remotely, nor, even more particularly, editing a graphical program, yet the Examiner continues to argue that Huntsman teaches editing a graphical program over a network.

Appellant respectfully reminds the Examiner that it is incumbent on the Examiner to particularly point out how the references teach Appellant's claimed subject matter, and that if the Examiner's citations do not teach the claimed features, which Appellant has shown above to indeed be the case, the Examiner should allow the claims.

Thus, Huntsman and Kodosky, taken singly or in combination, fail to teach or suggest this claimed feature of claim 59.

Nor does the cited art disclose **send information describing a user interface of the graphical program over the network to the client computer system after establishing the network connection with the client computer system, wherein the information regarding the user interface is useable by the client computer system to display the user interface on the client computer system**, as recited in claim 59.

Cited col.9:31-50, discussed above, describes sending GUI information for a conventional text-based program, *not* information describing a user interface *of a graphical program* as defined in the claim.

As explained above, per this citation, Huntsman's GIF file, as referenced by the REMOTE.HTM, presents the screen image of the client (first) computer, and REMOTE.HTM contains appropriate HTML references to the GIF file so that the GIF file will be displayed as a clickable image, and thus appears to be a GUI. As one of skill in the programming arts would readily understand, such a GUI is not itself a graphical program as defined in the claim, nor, more specifically, is Huntsman's GUI a user interface *for a graphical program*. As defined in claim 59, a graphical program includes the block diagram that includes a plurality of interconnected function icons representing graphical data flow of a desired function, and where executing the graphical program includes executing the block diagram.

As known by those of skill in the art, a textual program that provides graphics functionality is not a graphical program. Moreover, Huntsman never even mentions or hints at a graphical program. Appellant has made this point clearly in previous Responses, yet the Examiner continues to refer to Huntsman's program as a graphical program. Appellant respectfully submits that it is improper for the Examiner to attempt to redefine key terms in Appellant's claims in a manner contradictory to the definitions in the claims, and in contradiction to the Specification.

Thus, the cited art fails to disclose these features of claim 59.

Nor has a proper motivation to combine been provided. The suggested motivation: "because one of ordinary skill in the art would be motivated to remotely control block diagram information of a virtual instrument that is executing on one computer system from another computer system", simply states a presumed benefit of Appellant's invention as recited in claim 59, and appears to rely on hindsight analysis using claim 59 as a blueprint, which is improper.

The Advisory Action's assertion that it would be obvious "to modify Huntsman's MS-Windows program as a block diagram of a graphical program of a virtual instrument in order to allow the block diagram of the graphical program of the virtual instrument to

be remotely controlled by a user so that the user can have access to the block diagram information executing on another computer", is not a proper motivation to combine. First, note that "the block diagram information" of claim 59 is sent to the client (remote) system to display the block diagram, and does not execute on the local computer, contrary to the Advisory Action. Nor is Huntsman's system for remote control of textual-based programs changed substantially by modifying the programs to be graphical programs, since in Huntsman's system (even in combination with Kodosky) it is only the GUI information that is transmitted. The Examiner has attempted to modify Hunstman by simply adding the novel feature of claim 59 "send information regarding the block diagram of the graphical program over the network to the client computer system after establishing the network connection with the client computer system, wherein the information regarding the block diagram is useable by the client computer system to display the block diagram on a client computer system"–which is neither taught nor suggested by Hunstman or Kodosky, which is improper. Moreover, neither reference even hints at the desirability of sending block diagram information to a remote computer for display, and so the Examiner's inclusion of this feature is not only unsupported in the cited art, but is clearly based on hindsight analysis, which is also improper.

Appellant notes that neither reference mentions or even hints at the desirability or utility of such a combination. Thus, Huntsman and Kodosky are not properly combinable to make a prima facie case of obviousness.

Moreover, even were Huntsman and Kodosky properly combinable, which Appellant argues they are not, the resulting combination would still not produce Appellant's invention as expressed in claim 59, as explained in detail above.

Thus, for at least the reasons presented above, the cited art, taken singly or in combination, fails to teach or suggest all the features and limitations of claim 59, and so claim 59, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over the cited art, and are thus allowable.

Claims 65, 74:

In addition to the novel limitations of claims 59 and 64, the cited art fails to disclose **send information regarding the block diagram of the graphical program**

over the network to each of the plurality of client computer systems after establishing the network connection with each of the plurality of client computer systems, wherein the information regarding the block diagram is useable by each of the plurality of client computer systems to display the block diagram, as recited in claim 65.

Cited col.9:31-50 describes sending REMOTE.HTM with references to a GIF file that represents the screen/interface for the server computer, referring to it as information regarding a graphical program, which is incorrect, at least because REMOTE.HTM and GIF file relate to Huntsman's interface, not a block diagram. This text says nothing about a *graphical program* at all, much less sending information regarding a graphical program (nor, more specifically, a block diagram) to client computers over a network, nor, more particularly, where the information is usable by each client computer to display the graphical program or block diagram. A GUI is not a graphical program, as explained at length above, and thus Huntsman's sending of information for a GUI is not germane to sending information regarding a block diagram. Nor does combining Kodosky's block diagram with Huntsman's sending of GUI information remedy this deficiency–the combination still doesn't teach or suggest sending information regarding a block diagram over a network to a plurality of client computers for display on the client computers.

Thus, the cited art, taken singly or in combination, fails to teach or suggest the features and limitations of claim 65.

Claims 68, 75:

In addition to the novel limitations of claim 59, the cited art fails to disclose **wherein the information regarding the block diagram is useable by the client computer system to display the block diagram on the web browser**, as recited in claim 68.

The Examiner admits that Huntsman fails to disclose this feature, but asserts that Kodosky remedies this admitted deficiency, citing col.8:8-23. Applicant disagrees.

While Kodosky does disclose block diagrams, nowhere does Kodosky teach or suggest, or even hint at, (sending) information regarding a block diagram of a graphical program that is useable by a client computer system to display the block diagram on the

web browser. In fact, Kodosky nowhere even mentions displaying a block diagram in a web browser at all. Applicant notes that Huntsman nowhere discloses or even hints at displaying a block diagram (nor any kind of program source code) on a web browser of a client computer system.

Thus, the cited art, taken singly or in combination, fails to teach or suggest all the features and limitations of claim 68.

Claims 69, 78, 90, 100:

As a further example, the cited art also fails to disclose **receive user input specifying an edit to the block diagram from the client computer system over the network; and edit the block diagram according to the user input specifying the edit**, as recited in claim 69.

Cited col.9:50-57, 61-67, and col.10:1-6 describes a web browser determining mouse coordinates (and HTML mode variables) with respect to the presented GUI/screen, and sending these coordinates to the URL-addressed computer via hypertext-to-GUI-response means, but says nothing about a *block diagram* at all, much less receiving user input over the network editing a block diagram, and editing a block diagram accordingly. Nor does combining Kodosky's block diagram with Huntsman's sending of mouse coordinate and mode information remedy this deficiency, at least because Kodosky nowhere even hints at editing a block diagram remotely as claimed.

In the Response to Arguments, the Examiner argues that "Huntsman clearly discloses 'receiv[ing] user input specifying an edit to the graphical program from the client computer system over the network', citing Huntsman's remote GUI operations (col.10:1-20). However, Appellant notes that Huntsman never even mentions the term "edit", and nowhere describes editing any kind of program at all, much less a graphical program. Thus, Huntsman does not, and cannot, provide the feature of remote editing of a program, and so, even in combination with Kodosky, which only teaches *local* editing of a block diagram, fails to teach or suggest these features of claim 69.

Thus, the cited art, taken singly or in combination, fails to teach or suggest the features and limitations of claim 69.

For the foregoing reasons, it is respectfully submitted that the Examiner's rejection of claims 59-104 was erroneous, and reversal of the decision is respectfully requested.

The fee of $540.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Appellant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5150-38605/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2009-10-06 JCH/MSW

VIII. CLAIMS APPENDIX

The following lists claims 59-104, incorporating entered amendments, as on appeal.

59. A computer accessible memory medium that stores program instructions executable to:

establish a network connection with a client computer system over a network;

receive user input from the client computer system indicating a graphical program for execution;

execute the graphical program, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function, and wherein said executing the graphical program comprises executing the block diagram;

send information describing a user interface of the graphical program over the network to the client computer system after establishing the network connection with the client computer system, wherein the information regarding the user interface is useable by the client computer system to display the user interface on the client computer system; and

send information regarding the block diagram of the graphical program over the network to the client computer system after establishing the network connection with the client computer system, wherein the information regarding the block diagram is useable by the client computer system to display the block diagram on the client computer system;

wherein the user interface is operable to facilitate interaction between a user and the graphical program over the network.

60. The computer accessible memory medium of claim 59, wherein the program instructions are further executable to:

provide information indicating a plurality of graphical programs to the client computer system over the network, wherein the information indicating a plurality of

graphical programs is usable by the client computer system to display information indicating the plurality of graphical programs;

wherein, in indicating the graphical program for execution, the user input selects the graphical program from the plurality of graphical programs.

61. The computer accessible memory medium of claim 59, wherein the program instructions are further executable to:

receive user input to the graphical program from the client computer system over the network; and

provide the user input to the graphical program;

wherein the graphical program is operable to respond to the user input.

62. The computer accessible memory medium of claim 59,

wherein the graphical program produces a first output state; and

wherein said sending information describing a user interface of the graphical program comprises sending information indicative of the first output state.

63. The computer accessible memory medium of claim 62,

wherein the graphical program produces a second output state after the graphical program produces the first output state; and

wherein the program instructions are further executable to send a user interface update indicating the second output state to the client computer system.

64. The computer accessible memory medium of claim 59, wherein the program instructions are further executable to:

establish a network connection with each of a plurality of client computer systems; and

send information describing a user interface of the graphical program over the network to each of the plurality of client computer systems after establishing the network connection with each of the plurality of client computer systems.

65. The computer accessible memory medium of claim 64, wherein the program instructions are further executable to:

send information regarding the block diagram of the graphical program over the network to each of the plurality of client computer systems after establishing the network connection with each of the plurality of client computer systems, wherein the information regarding the block diagram is useable by each of the plurality of client computer systems to display the block diagram.

66. The computer accessible memory medium of claim 59,

wherein the graphical program executes to perform a measurement or automation function.

67. The computer accessible memory medium of claim 59, wherein the network is the Internet.

68. The computer accessible memory medium of claim 59,

wherein the information describing the user interface is useable by the client computer system to display the user interface of the graphical program on a web browser; and

wherein the information regarding the block diagram is useable by the client computer system to display the block diagram on the web browser.

69. The computer accessible memory medium of claim 59, wherein the program instructions are further executable to:

receive user input specifying an edit to the block diagram from the client computer system over the network; and

edit the block diagram according to the user input specifying the edit.

70. The computer accessible memory medium of claim 59,

wherein the user interface of the graphical program comprises at least one input variable icon for providing inputs to the block diagram and at least one output variable icon for displaying outputs produced by the block diagram.

71. The computer accessible memory medium of claim 59, wherein the program instructions are further executable to:

receive input of at least one input variable from the client computer system over the network;

the block diagram executing using the input of the at least one input variable;

the block diagram generating an output of at least one output variable; and

providing the output of the at least one output variable to the client computer system over the network for display.

72. The computer accessible memory medium of claim 59,

wherein the graphical program implements a virtual instrument; and

wherein the user interface of the graphical program comprises a front panel of the virtual instrument.

73. A method for executing a graphical program on a first computer and providing a user interface of the graphical program for display on a second computer, the method comprising:

executing the graphical program on the first computer, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function, wherein the first computer and the second computer are connected over a network, and wherein said executing the graphical program comprises executing the block diagram;

providing information describing the user interface of the graphical program to the second computer during said executing, wherein said providing comprises the first

computer providing the information describing the user interface of the graphical program over the network to the second computer, and wherein the information describing the user interface is useable by the second computer to display the user interface of the graphical program on the second computer; and

providing information regarding the block diagram of the graphical program to the second computer over the network, wherein said providing comprises the first computer providing the information regarding the block diagram of the graphical program over the network to the second computer, wherein the information regarding the block diagram is useable by the second computer to display the block diagram on the second computer;

wherein the user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer.

74. The method of claim 73, further comprising:

providing information describing the user interface of the graphical program to a plurality of computers over the network during said executing, where the information describing the user interface of the graphical program is useable by each of the plurality of computers to display the user interface of the graphical program.

75. The method of claim 73,

wherein the information describing the user interface is useable by the second computer to display the user interface of the graphical program on a web browser of the second computer; and

wherein the information regarding the block diagram is useable by the second computer to display the block diagram on the web browser of the second computer.

76. The method of claim 73, further comprising:

the graphical program executing on the first computer responding to user input received to the graphical program via the displayed user interface on the second computer; wherein the user input is provided to the first computer over the network.

77. The method of claim 73, wherein the graphical program produces a second output state after the graphical program produces a first output state, the method further comprising:

providing a user interface update over the network indicating the second output state, where the user interface update is useable by the second computer to update the user interface displayed on the second computer.

78. The method of claim 73, further comprising:

receiving user input specifying an edit to the block diagram to the first computer from the second computer over the network; and

editing the block diagram according to the user input specifying the edit, wherein said editing is performed by the first computer.

79. The method of claim 73, wherein the user interface of the graphical program comprises at least one input variable icon for providing inputs to the block diagram and at least one output variable icon for displaying outputs produced by the block diagram, the method further comprising:

receiving input of at least one input variable to the first computer from the second computer over the network;

the block diagram executing using the input of the at least one input variable;

the block diagram generating an output of at least one output variable; and

providing the output of the at least one output variable to the second computer over the network, wherein the output is displayable on the second computer.

80. The method of claim 73,

wherein the graphical program implements a virtual instrument; and

wherein the user interface of the graphical program comprises a front panel of the virtual instrument.

81. A system for executing a graphical program, the system comprising:

a first computer, comprising:

a processor; and

a memory, coupled to the processor;

wherein the first computer is operable to couple to a network;

wherein the memory stores a graphical program, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function;

wherein the first computer is operable to execute the graphical program and provide information describing a user interface of the graphical program over the network to a second computer during said executing, wherein said executing the graphical program comprises executing the block diagram;

wherein the information describing the user interface over the network is useable by the second computer to display the user interface of the graphical program;

wherein the user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer; and

wherein the first computer is operable to provide information regarding the block diagram of the graphical program over the network to the second computer, wherein the information regarding the block diagram is useable by the second computer to display the block diagram on the second computer.

82. A computer accessible memory medium that stores program instructions executable to:

receive user input at a first computer indicating a graphical program, wherein the graphical program is stored on a server computer, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function;

provide the user input indicating the graphical program over a network to the server computer;

receive information describing a user interface of the graphical program from the server computer over the network during execution of the graphical program on the server computer;

receive information regarding the block diagram of the graphical program from the server computer over the network;

display the user interface of the graphical program at the first computer based on the information describing a user interface; and

display the block diagram at the first computer based on the information regarding the block diagram;

wherein the user interface facilitates interaction between a user and the graphical program executing on the server computer.

83. The computer accessible memory medium of claim 82,

wherein the graphical program executes to perform a measurement or automation function.

84. The computer accessible memory medium of claim 82, wherein the program instructions are further executable to:

establish a network connection with the server computer over the network after said receiving user input indicating the graphical program;

wherein said receiving information describing the user interface and said receiving information regarding the block diagram are performed after said user input indicating the graphical program and after said establishing a network connection.

85. The computer accessible memory medium of claim 84,

wherein the graphical program is already executing on the server computer when said establishing a network connection occurs.

86. The computer accessible memory medium of claim 82,

wherein to display the user interface of the graphical program, the program instructions are executable to display the user interface of the graphical program on a web browser.

87. The computer accessible memory medium of claim 82, wherein the program instructions are further executable to:

receive user input to the graphical program via the displayed user interface; and

provide the user input to the server computer over the network for input to the graphical program executing on the server computer.

88. The computer accessible memory medium of claim 82,

wherein the graphical program produces a first output state; and

wherein said displaying the user interface includes displaying the user interface illustrating the first output state.

89. The computer accessible memory medium of claim 82, wherein the graphical program produces a second output state after the graphical program produces a first output state, wherein the program instructions are further executable to:

receive a user interface update over the network indicating the second output state; and

update the user interface in response to the user interface update.

90. The computer accessible memory medium of claim 82, wherein the program instructions are further executable to:

receive user input specifying an edit to the block diagram; and

provide the user input specifying the edit to the server computer over the network;

wherein the first computer is operable to edit the block diagram according to the user input specifying the edit.

91. The computer accessible memory medium of claim 82,

wherein said indicating the graphical program comprises providing a uniform resource locator (URL).

92. The computer accessible memory medium of claim 82, wherein the program instructions are further executable to:

display information indicating a plurality of graphical programs on the first computer;

wherein, in indicating the graphical program on the first computer, the user input selects the graphical program from the plurality of graphical programs.

93. The computer accessible memory medium of claim 82,

wherein the user interface of the graphical program comprises at least one input variable icon for providing inputs to the block diagram and at least one output variable icon for displaying outputs produced by the block diagram.

94. The computer accessible memory medium of claim 82, wherein the program instructions are further executable to:

receive user input manipulating input of at least one input variable;

provide the user input manipulating input of the at least one input variable to the server computer over the network;

receive output of at least one output variable from the server computer over the network, wherein the output is generated by the block diagram executing using the manipulated input of the at least one input variable; and

display the output of the at least one output variable.

95. The computer accessible memory medium of claim 82,

wherein the graphical program implements a virtual instrument; and

wherein the user interface of the graphical program comprises a front panel of the virtual instrument.

96. A method for displaying a graphical user interface and block diagram of a graphical program on a second computer in response to execution of the graphical program on a first computer, the method comprising:

receiving user input to the second computer, wherein the user input indicates the graphical program on the first computer, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function, wherein the first computer and the second computer are connected over a network;

receiving information describing a graphical user interface of the graphical program at the second computer from the first computer over the network during execution of the graphical program on the first computer;

receiving information regarding the block diagram of the graphical program at the second computer from the first computer over the network;

displaying the graphical user interface of the graphical program on the second computer based on the information describing the graphical user interface; and

displaying the block diagram on the second computer, using the information regarding the block diagram;

wherein the graphical user interface facilitates interaction between a user of the second computer and the graphical program executing on the first computer.

97. The method of claim 96, further comprising:

establishing a network connection with the first computer over the network after said receiving user input indicating the graphical program;

wherein said receiving information describing the graphical user interface and said receiving information regarding the block diagram are performed after said user input indicating the graphical program and after said establishing a network connection.

98. The method of claim 96,

wherein said displaying the graphical user interface of the graphical program comprises displaying the graphical user interface of the graphical program on a web browser.

99. The method of claim 96, further comprising:

receiving user input to the graphical program via the displayed graphical user interface; and

providing the user input to the first computer over the network for input to the graphical program executing on the first computer.

100. The method of claim 96, further comprising:

receiving user input specifying an edit to the block diagram; and

providing the user input specifying the edit to the first computer over the network;

wherein the user input specifying the edit is useable by the first computer to edit the graphical program.

101. The method of claim 96,

wherein the graphical user interface of the graphical program comprises at least one input variable icon for providing inputs to the block diagram and at least one output variable icon for displaying outputs produced by the block diagram.

102. The method of claim 96, further comprising:

receiving user input manipulating input of at least one input variable;

providing the user input manipulating input of the at least one input variable to the first computer over the network;

receiving output of at least one output variable from the first computer, wherein the output is generated by the block diagram executing using the manipulated input of the at least one input variable; and

displaying the output of at least one output variable on the second computer.

103. The method of claim 96,

wherein the graphical program implements a virtual instrument; and

wherein of the graphical program comprises a front panel of the virtual instrument.

104. A system, comprising:

a processor; and

a memory, coupled to the processor;

a network port operable to couple to a network;

wherein the memory stores program instructions executable by the processor to:

receive user input specifying a graphical program, wherein the graphical program includes a block diagram that comprises a plurality of interconnected function icons representing graphical data flow of a desired function;

provide the user input specifying the graphical program over a network to a server computer;

receive information describing a user interface of the graphical program from the server computer over the network during execution of the graphical program on the server computer;

receive information regarding the block diagram of the graphical program from the server computer over the network;

display the user interface of the graphical program based on the information describing a user interface; and

display the block diagram based on the information regarding the block diagram;

wherein the user interface facilitates interaction between a user and the graphical program executing on the server computer.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.